Extracting News Information from the Web with Machine Learning

Degree programme: BSc in Computer Science | Specialisation: Computer Perception and Virtual Reality
Thesis advisor: Prof. Dr. Jürgen Vogel
Expert: Andreas Dürsteler (Swisscom)

The web provides an abundance of structured information. Extracting the desired content can be very tedious though, especially if your goal is to extract it from many websites about which you don't know much in advance. The proposed solution uses machine learning to find predefined items from the HTML source code and store them in a database.

Introduction

An increasing number of newspapers are going out of print, either for good or to hereafter only have an online presence. At the same time, the internet is flooded with self-proclaimed «news networks» and all sorts of websites producing «news content». The more content there is, the more could be analyzed automatically.
The focus of this thesis was to create a tool able to find and extract desired items from news websites (e.g. teasers, news articles, comments etc.) and store them in a relational database for later analysis.

Method

The information we want to extract is located inside certain tags within the HTML source code of each website. These tags are themselves nested within other tags and they may have siblings or children. For my purpose, I have treated this HTML structure like a tree and each tag like a node. This way, I could use the nodes as targets for classifying while using their absolute xPath as unique identifier.
The main idea was to divide the item extraction problem into sub-problems and solve each of them sequentially. First, I do a depth first traversal of the whole web page, extracting features for each node while propagating descendants' information upwards. Then, I use a classifier to find container nodes for each item found in the source code (a container node is the topmost node of a sub-tree containing the whole item, see figure 2). Then, for each container node, another classifier is used to find the nodes containing item attributes (e.g. title, text, author etc.) within each subtree. Finally, the information is extracted from each identified node and stored in the database.
The data set used to train the classifiers was created manually using the Firefox browser, its «Inspect element» function and a browser extension called «xPath Finder».

Conclusion

The proposed method is an interesting take on an unsolved problem. Most alternatives in literature propose rule-based solutions, which are fast and provide good results, but they are at the same time very specific. My approach can be generalized and adapted, provided there is a prepared data set for training.
I think that, with a large enough data set and the right features, this method can lead to a powerful tool in information extraction.

Michel Hosmann
hosmann@gmx.net

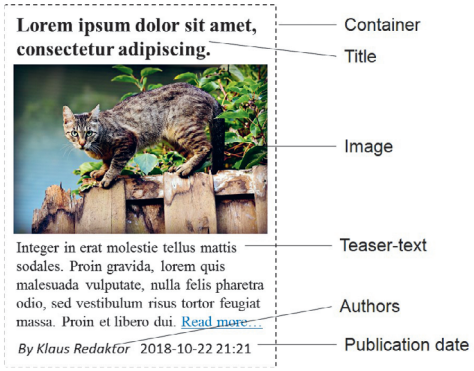

Figure 1: A news teaser with our target elements.

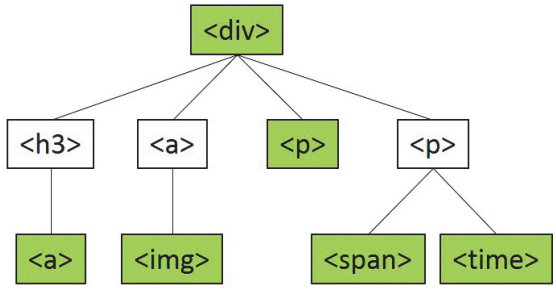

Figure 2: Example HTML sub-tree for the teaser in fig. 1, target nodes in green.